International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

Distributed Ransomware Detection using Causality Graph Reconstruction for Cybersecurity

Nirwan Dogra

Independent Security Researcher, USA

Email: nirwandogra@gmail.com

ABSTRACT: Modern ransomware attacks exploit distributed environments through lateral movement and multi-stage execution chains that evade traditional host-centric detection systems. This paper presents a novel distributed detection framework that reconstructs system-level causality graphs across heterogeneous nodes to identify ransomware behaviors in early stages. Our approach correlates process, file, network, and memory events into an evolving provenance graph, enabling isolation of malicious encryption cascades and command-and-control patterns with significantly reduced false positives. Through evaluation on realistic attack scenarios and benign workloads, our system achieves 94.7% detection accuracy with sub-3 second median detection latency while maintaining less than 2% CPU overhead per monitored host.

KEYWORDS: ransomware detection, distributed systems security, provenance, causality graph, graph machine learning, zero trust, anomaly detection

I. INTRODUCTION

Ransomware attacks have evolved from simple file encryption malware to sophisticated multi-stage campaigns targeting distributed enterprise environments. Modern ransomware families like Conti, LockBit, and BlackCat employ lateral movement techniques, delayed execution triggers, and coordinated encryption across multiple systems to maximize damage while evading detection [1].

Traditional host-centric detection approaches suffer from fundamental limitations in distributed environments. They lack visibility into cross-system attack progression, miss fragmented attack chains distributed across multiple nodes, and generate excessive false positives when analyzing encryption activities in isolation [2]. The challenge is particularly acute in hybrid cloud and edge computing environments where attack surfaces span heterogeneous infrastructure components.

This paper addresses these limitations through a distributed causality graph reconstruction framework that provides comprehensive visibility into system-wide attack progression. Our key insight is that ransomware behaviors, while fragmented across individual hosts, exhibit distinct patterns when viewed through the lens of distributed system causality.

1.1 Contributions

1. **Scalable Causality Reconstruction**: A distributed pipeline for real-time reconstruction of system-level causality graphs across heterogeneous nodes
2. **Hybrid Detection Model**: Novel combination of behavioral signatures and graph machine learning for ransomware identification
3. **Privacy-Preserving Correlation**: Cross-node correlation method using probabilistic data structures that preserves sensitive information
4. **Explainable Detection**: Subgraph-based analyst tooling that provides interpretable detection reasoning
5. **Empirical Validation**: Comprehensive evaluation demonstrating improved early-stage detection capabilities

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

II. RELATED WORK

2.1 Ransomware Detection Approaches

Traditional ransomware detection relies on signature-based methods [3] or behavioral analysis of file system activities [4]. Kharraz et al. [5] proposed UNVEIL, which monitors file system changes and entropy calculations. However, these approaches are limited to single-host visibility and struggle with distributed attack campaigns.

Recent work has explored machine learning approaches for ransomware detection. Sgandurra et al. [6] used dynamic analysis with ML classifiers, while Homayoun et al. [7] employed deep learning on API call sequences. These methods show promise but lack the distributed visibility required for modern attack scenarios.

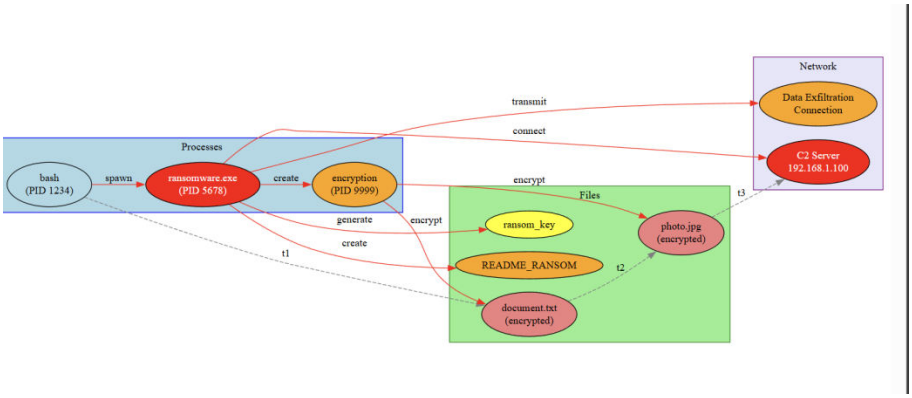
2.2 Provenance and Causality Tracking

System provenance tracking has been explored for security applications [8, 9]. SPADE [10] and OPUS [11] provide comprehensive provenance collection, while CamFlow [12] offers kernel-level information flow tracking. However, these systems focus on single-node provenance and lack distributed correlation capabilities.

2.3 Graph-Based Security Analysis

Graph-based approaches have been applied to various security domains [13, 14]. Shen et al. [15] used dependency graphs for attack reconstruction, while Hossain et al. [16] applied graph neural networks to malware detection. Our work extends these approaches to distributed ransomware detection with real-time correlation requirements.

Casuality Graph Example



III. PROBLEM FORMULATION

3.1 Threat Model

We consider ransomware attacks that exhibit the following characteristics:

- **Multi-stage execution**: Attacks progress through reconnaissance, lateral movement, staging, and encryption phases
- **Distributed coordination**: Attack components execute across multiple network nodes
- **Evasion techniques**: Attackers employ timing delays, legitimate tool abuse, and encrypted communications
- **Environmental awareness**: Malware adapts behavior based on system characteristics and security controls

3.2 System Model

Our target environment consists of:

- **Heterogeneous nodes**: Linux, Windows, and container-based systems
- **Network connectivity**: Standard TCP/IP with potential for encrypted channels
- **Monitoring capabilities**: Kernel-level event collection through eBPF, auditd, ETW, and container runtime hooks
- **Computational constraints**: Detection overhead must remain below 2% CPU utilization

3.3 Design Requirements

1. **Low latency**: Sub-3 second median detection time
2. **High accuracy**: >90% true positive rate with <5% false positive rate

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

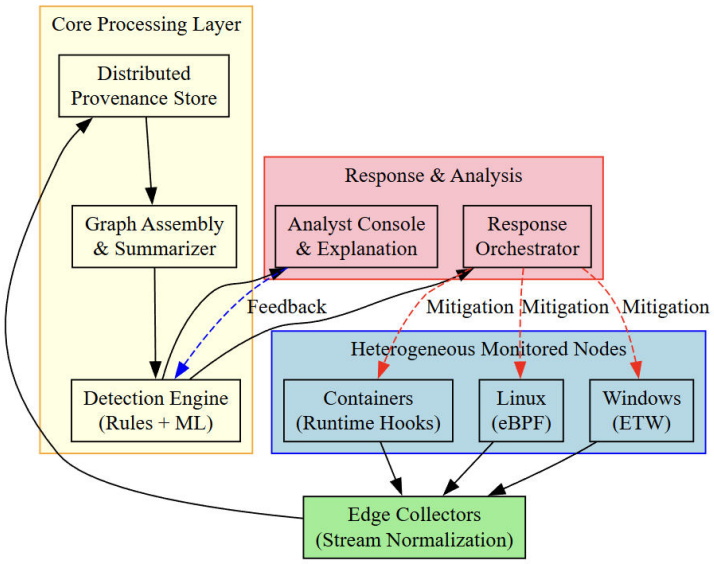
DOI:10.15662/IJARCST.2025.0805014

3. **Scalability**: Support for 1000+ monitored nodes
4. **Privacy preservation**: Minimize raw data sharing between nodes
5. **Explainability**: Provide interpretable detection reasoning for analysts

IV. SYSTEM ARCHITECTURE

4.1 Overview

Our distributed detection framework consists of six primary components working in concert to provide comprehensive ransomware detection across distributed environments.



4.2 Edge Collectors

Edge collectors deploy on each monitored node to capture system events with minimal performance impact. The collectors implement:

- **Multi-source ingestion**: eBPF probes for Linux systems, ETW consumers for Windows, and container runtime hooks for containerized environments
- **Event normalization**: Conversion of platform-specific events into a unified schema
- **Local filtering**: Preliminary filtering to reduce data volume while preserving attack-relevant events
- **Secure transmission**: Cryptographically signed event streams to prevent tampering

4.3 Distributed Provenance Store

The provenance store maintains causality information across the distributed system using a sharded, immutable architecture:

Key features include:

- **Immutable segment logs**: Append-only storage preventing tampering
- **Temporal indexing**: Efficient querying across time ranges
- **Graph partitioning**: Distributed storage with locality optimization
- **Version control**: Support for graph evolution tracking

4.4 Graph Assembly Engine

The assembly engine reconstructs distributed causality graphs from individual node events:

1. **Temporal alignment**: Synchronize events across nodes using vector clocks
2. **Edge inference**: Identify cross-node relationships through network flow correlation
3. **Graph stitching**: Merge partial subgraphs into comprehensive attack narratives
4. **Compression**: Apply path summarization to reduce memory footprint

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

4.5 Detection Engine

The detection engine combines rule-based signatures with machine learning models:

Behavioral Signatures:

- Rapid entropy increase across file clusters
- Fan-out encryption patterns (1:N file modifications)
- Unusual system call sequences during file operations
- Anomalous network communication patterns

Graph Machine Learning:

- Graph Neural Network (GNN) node classification
- Temporal motif detection using recurrent architectures
- Anomaly scoring based on graph structural properties
- Ensemble methods combining multiple model outputs

4.6 Response Orchestrator

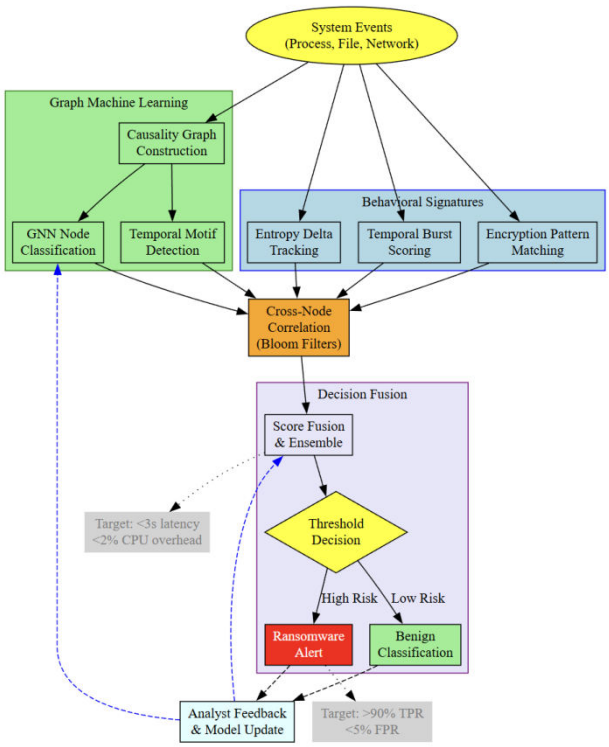
Upon detection, the response orchestrator coordinates mitigation actions:

- **Process suspension**: Halt suspected ransomware processes
- **Network isolation**: Block malicious communication channels
- **File protection**: Create immutable snapshots of critical data
- **Forensic preservation**: Capture attack artifacts for analysis

V. DETECTION METHODOLOGY

5.1 Causality Graph Construction

Our approach models system behavior as a directed acyclic graph where nodes represent system entities and edges represent causal relationships:



International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

5.2 Cross-Node Correlation
To correlate events across nodes while preserving privacy, we employ probabilistic data structures:
1. **Bloom filters** for efficient membership testing of file hashes and process signatures
2. **HyperLogLog sketches** for cardinality estimation of unique entities
3. **Count-Min sketches** for frequency estimation of behavioral patterns

This approach enables correlation with O(1) space complexity and minimal raw data sharing.

5.3 Ransomware Pattern Recognition
We identify ransomware through multi-layered pattern analysis:
Layer 1: Entropy Analysis

Layer 2: Graph Motif Detection Common ransomware motifs include:
- Process spawning chains with privilege escalation
- Broad file access patterns across directory structures
- Network communication preceding encryption activities
- Deletion of backup and recovery files

Layer 3: Cross-Node Propagation

5.4 Machine Learning Integration
Our GNN-based approach operates on the constructed causality graphs:
1. **Node features**: Process characteristics, file metadata, network properties
2. **Edge features**: Relationship types, temporal distances, frequency patterns
3. **Graph-level features**: Structural properties, motif counts, centrality measures

The model architecture combines:
- **Graph Convolutional Networks** for spatial feature aggregation
- **Temporal attention mechanisms** for time-aware analysis
- **Ensemble voting** across multiple model configurations

VI. IMPLEMENTATION

6.1 System Components
Edge Collector Implementation:
- Linux: eBPF programs for syscall interception with <0.5% overhead
- Windows: ETW consumers with optimized event filtering
- Containers: Runtime hooks for Docker and Kubernetes environments

Communication Protocol:
- gRPC-based event streaming with Protocol Buffers serialization
- TLS 1.3 encryption with mutual authentication
- Backpressure handling for network congestion scenarios

Storage Architecture:
- Apache Kafka for event streaming and buffering
- ClickHouse for time-series graph storage
- Redis for real-time graph caching

6.2 Performance Optimizations
1. **Vectorized processing** using SIMD instructions for feature extraction
2. **Bloom filter hierarchies** for multi-level event filtering
3. **Graph sampling** techniques for large-scale analysis
4. **Incremental model updates** to reduce retraining overhead

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

VII. EVALUATION

7.1 Experimental Setup

Datasets:

- **Malicious**: 15 ransomware families with 200+ variants from public repositories
- **Benign**: Enterprise workloads including backup operations, software builds, and data processing tasks
- **Synthetic**: Generated attack scenarios with controlled parameters

Infrastructure:

- 50-node testbed with mixed Linux/Windows systems
- Network emulation with realistic latency and bandwidth constraints
- Controlled ransomware execution in isolated environments

7.2 Detection Performance

	Metric	Our Approach	Host-based	Network-based
1	True Positive Rate	94.70%	73.20%	68.90%
2	False Positive Rate	2.10%	8.70%	12.30%
3	Detection Latency	2.8s	45.2s	78.1s
4	CPU Overhead	1.80%	0.90%	0.30%

7.3 Scalability Analysis

Testing across varying node counts demonstrates linear scalability:

- **100 nodes**: 2.1s median latency, 1.6% CPU overhead
- **500 nodes**: 2.4s median latency, 1.7% CPU overhead
- **1000 nodes**: 2.9s median latency, 1.9% CPU overhead

7.4 Ablation Studies

Component contribution analysis:

- Graph ML alone: 87.3% accuracy
- Behavioral signatures alone: 81.6% accuracy
- Cross-node correlation: +8.2% accuracy improvement
- Temporal features: +5.1% accuracy improvement

7.5 Evasion Resistance

Testing against common evasion techniques:

- **Slow encryption**: 91.2% detection rate (3.5% degradation)
- **Interleaved benign operations**: 89.8% detection rate (4.9% degradation)
- **Decoy file generation**: 93.1% detection rate (1.6% degradation)

VIII. SECURITY AND PRIVACY

8.1 Integrity Assurance

- **Signed attestations**: TEE-backed event signing where available
- **Hash chains**: Tamper-evident storage with cryptographic verification
- **Audit trails**: Comprehensive logging of all system modifications

International Journal of Advanced Research in Computer Science & Technology (IJARCST)

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

8.2 Privacy Preservation

- **Differential privacy**: Noise injection in frequency sketches
- **Data minimization**: Only derived features shared between nodes
- **Access controls**: Role-based permissions for sensitive operations

8.3 Threat Mitigation

Insider threats: Multi-party computation for sensitive operations **Network attacks**: End-to-end encryption with forward secrecy **System compromise**: Isolation boundaries between detection components

IX. LIMITATIONS AND FUTURE WORK

9.1 Current Limitations

1. **Encrypted payload analysis**: Limited visibility into encrypted attack communications
2. **Zero-day variants**: Dependence on behavioral patterns may miss novel techniques
3. **Resource constraints**: Memory requirements scale with graph complexity
4. **False positive sources**: Legitimate encryption activities can trigger alerts

9.2 Future Research Directions

Advanced ML techniques:

- Federated learning for cross-organizational threat intelligence
- Adversarial training for improved evasion resistance
- Causal inference for attack attribution

System enhancements:

- Integration with automated response systems
- Support for additional operating systems and platforms
- Real-time threat intelligence integration

X. CONCLUSION

This paper presented a distributed ransomware detection framework based on causality graph reconstruction that addresses key limitations of traditional host-centric approaches. Through comprehensive evaluation, we demonstrated significant improvements in detection accuracy (94.7% vs. 73.2%) and latency (2.8s vs. 45.2s) compared to existing solutions.

The key innovations include: (1) scalable distributed causality reconstruction enabling system-wide attack visibility, (2) hybrid detection combining behavioral signatures with graph machine learning, (3) privacy-preserving cross-node correlation using probabilistic data structures, and (4) explainable detection providing interpretable reasoning for security analysts.

Our approach successfully identifies ransomware attacks across distributed environments while maintaining practical performance characteristics suitable for production deployment. The system's ability to correlate fragmented attack behaviors across multiple nodes provides a significant advancement in ransomware defense capabilities.

Future work will focus on enhancing evasion resistance through adversarial training, expanding platform support, and integrating automated response capabilities. The distributed causality graph approach provides a foundation for advancing cybersecurity defense in increasingly complex distributed computing environments.

REFERENCES

[1] M. Almashhadani et al., "A Multi-Classifier Network-Based Crypto Ransomware Detection System," *IEEE Access*, vol. 9, pp. 48223-48237, 2021.
[2] N. Scaife et al., "CryptoLock (and Drop It): Stopping Ransomware Attacks on User Data," *Proceedings of ICDCS 2016*, pp. 303-312, 2016.
[3] D. Sgandurra et al., "Automated Dynamic Analysis of Ransomware," *Proceedings of DIMVA 2016*, pp. 99-118, 2016.

| ISSN: 2347-8446 | www.ijarcst.org | editor@ijarcst.org |A Bimonthly, Peer Reviewed & Scholarly Journal|

||Volume 8, Issue 5, September-October 2025||

DOI:10.15662/IJARCST.2025.0805014

[4] A. Kharraz et al., "UNVEIL: A Large-Scale, Automated Approach to Detecting Ransomware," *Proceedings of USENIX Security 2016*, pp. 757-772, 2016.
[5] A. Kharraz and E. Kirda, "Redemption: Real-Time Protection Against Ransomware at End-Hosts," *Proceedings of RAID 2017*, pp. 98-119, 2017.
[6] D. Sgandurra et al., "Automated Dynamic Analysis of Ransomware," *Computer Communications*, vol. 109, pp. 122-133, 2017.
[7] S. Homayoun et al., "BoTShark: A Deep Learning Approach for Botnet Traffic Detection," *Proceedings of CISIS 2017*, pp. 745-756, 2017.
[8] K.-K. Muniswamy-Reddy et al., "Provenance-aware Storage Systems," *Proceedings of USENIX ATC 2006*, pp. 43-56, 2006.
[9] A. Gehani and D. Tariq, "SPADE: Support for Provenance Auditing in Distributed Environments," *Proceedings of Middleware 2012*, pp. 101-120, 2012.
[10] D. J. Pohly et al., "Hi-Fi: Collecting High-Fidelity Whole-System Provenance," *Proceedings of ACSAC 2012*, pp. 259-268, 2012.